LQueue.h and Lqueue.cpp is taken from previous project

**StaQueue.h file:**

#pragma once

#ifndef STACK\_H

#define STACK\_H

#include "LQueue.h"

class StaQueue {

private:

LQueue q1;

LQueue q2;

public:

StaQueue();

~StaQueue();

void push(int item);

void pop();

int top();

bool empty();

friend ostream& operator <<(ostream&, StaQueue&);

};

#endif

**StaQueue.cpp file:**

#include "Stack.h"

StaQueue::StaQueue() {}

StaQueue::~StaQueue() {}

void StaQueue::push(int item) {

while (!q1.isEmpty()) {

q2.Enqueue(q1.front());

q1.Dequeue();

}

q1.Enqueue(item);

while (!q2.isEmpty()) {

q1.Enqueue(q2.front());

q2.Dequeue();

}

}

void StaQueue::pop() {

if (empty()) {

cout << "Stack is empty!" << endl;

return;

}

q1.Dequeue();

}

int StaQueue::top() {

if (empty()) {

cout << "Stack is empty!" << endl;

return -1;

}

return q1.front();

}

bool StaQueue::empty() {

return q1.isEmpty();

}

ostream& operator<<(ostream& out, StaQueue& S) {

out << "Stack: " << S.q1;

return out;

}

**Main.cpp file:**

#include<iostream>

#include"LQueue.h"

#include"Stack.h"

using namespace std;

int main() {

StaQueue SQ;

for (int i = 0; i < 8; i++) {

SQ.push(i);

cout << SQ << endl;

cout << SQ.top() << endl;

}

for (int i = 0; i < 4; i++) {

SQ.pop();

cout << SQ << endl;

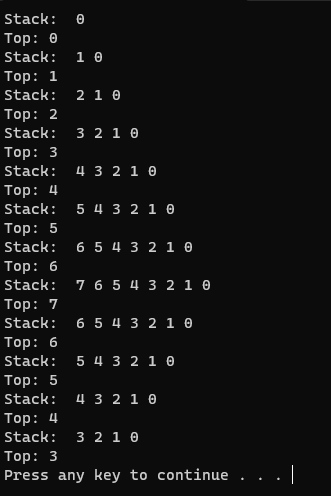
cout << SQ.top() << endl;

}

system("pause");

return 0;

}

****